2.2. TAD TRIE (Retrieval)

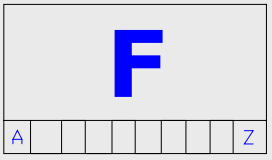
DEFINICIONES

- **Árbol para representar conjuntos de cadenas de caracteres u objetos (DICCIONARIO DE CADENAS):**
  - Cada nodo representa el prefijo de una palabra
  - Los hijos de un nodo representan las cadenas que tiene a sus padres como prefijos
- **VENTAJA: Realizar búsquedas parciales (palabras que empiezan por "AR")**

2.2. TAD TRIE (Retrieval)

IMPLEMENTACION

- **Nodo:**
  - Campo booleano (indica si es una palabra completa o un prefijo)
  - Estructura de punteros para todos los posibles hijos



2.2. TAD TRIE (Retrieval)

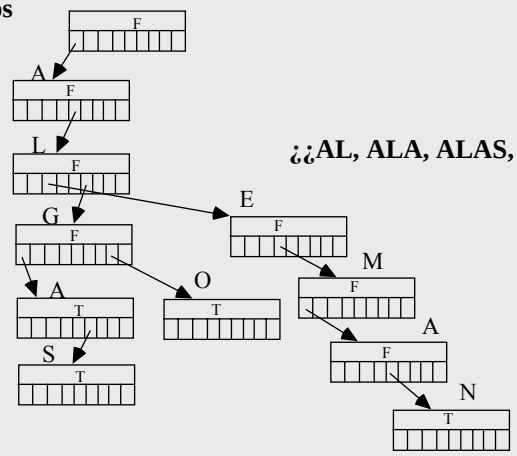
FUNCIONES (I)

- ***Crear:***

  Constructor que pondrá la variable booleana a FALSE y los punteros nulos

ALGA, ALGAS, ALGO, ALEMAN

¿¿AL, ALA, ALAS, ALE??



2.2. TAD TRIE (Retrieval)

FUNCIONES (III)

- ***Pertenece:***

```
ALGORITMO BUSQUEDA
        ENTRADA: s : Cadena; A : TRIE;
        SALIDA: Encontrado: Boolean;
        VAR :    p: Iterador; c: Carácter;
    METODO
        p = A; Encontrado = FALSE
        Mientras NO EsVacio (p) y NO Encontrado
                Si LONGITUD (s) == 0
                    Encontrado = Dato (p)
                Sino
                    c = OBTENER (s,1)
                    p = p.HIJO (c)
                    Si NO EsVacio (p)
                            SUPRIMIR (s)
                    fsi
                fsi
        fMientras
    fMETODO
```

2.2. TAD TRIE (Retrieval)

FUNCIONES (IV)

- ***Inserción:***

  • ***Si el camino seguido para la inserción ya existe:***

  Cambiar el flag booleano a cierto al llegar a la última letra de la palabra

  • ***Si no existe:***

  Crear un nodo en la posición que le corresponda, y así con todas las letras hasta completar la palabra completa

2.2. TAD TRIE (Retrieval)

COMPLEJIDAD (I)

- ***Complejidad Temporal:***

  PERTENECE e INSERTAR:

  TRIE: O (L) *L = longitud de la cadena*

  ABB: O (n) *n = tamaño diccionario*

2.2. TAD TRIE (Retrieval)

COMPLEJIDAD (II)

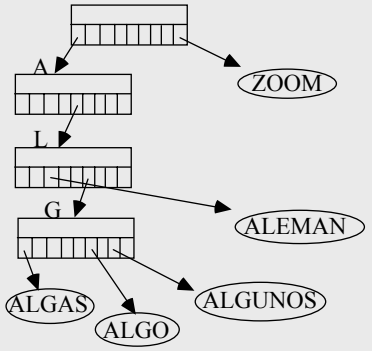
- ***Complejidad Espacial:***

  Demasiados nodos. Otra opción es tener dos tipos de nodo:

  - *Nodos prefijo.*
  - *Nodos terminales.*

ALGAS, ALGO, ALGUNOS, ALEMAN, ZOOM

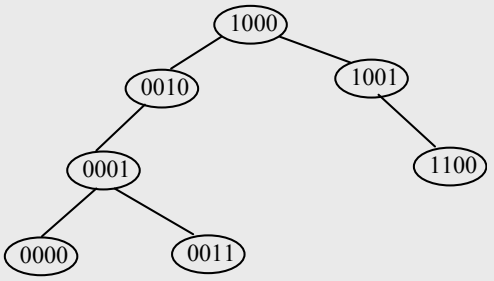
¿¿AL, ALA, ZORRO??



2.3. ARB. BÚSQUEDA DIGITALES

DEFINICIÓN

- *Caso particular de los TRIES*
- *Árbol binario en que cada nodo contiene un elemento.*
- *La asignación de un nodo viene determinada por la representación binaria de la clave:*
  - ***Dado nodo X en nivel "i":***
    - ***∀ Y ∈ SubárbolIzq (X) tienen el bit "i" igual a cero***
    - ***∀ Y ∈ SubárbolDer (X) tienen el bit "i" igual a uno***



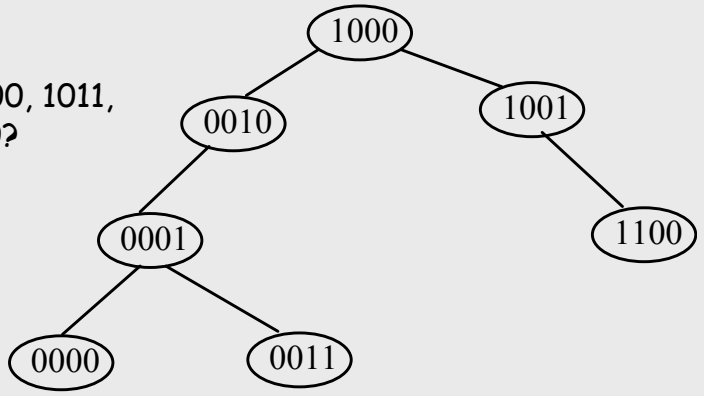
2.3. ARB. BÚSQUEDA DIGITALES

OPERACIONES (I)

- ***BUSQUEDA, INSERCION Y BORRADO:***

Igual que en ABB excepto que el subárbol al que hay que moverse viene determinado por un bit en la clave.

¿0100, 1011, 1010?



2.3. ARB. BÚSQUEDA DIGITALES

COMPLEJIDAD

- ***COMPLEJIDAD:***

O(h) *h = altura del árbol*

h = N+1 *N = nº de bits de la clave*